

Spring Boot WebSocket

实践作业需求

- 搭建一个websocket服务端
- 搭建一个websocket客户端

WebSocket

WebSocket 协议是基于 TCP 的一种网络协议，它实现了浏览器与服务器全双工（Full-duplex）通信。它的最大特点就是，服务器可以主动向客户端推送信息，客户端也可以主动向服务器发送信息，是真正的双向平等对话，属于服务器推送技术的一种。在 WebSocket API 中，浏览器和服务器只需要完成一次握手，两者之间就直接可以创建持久性的连接，并进行双向数据传输。

Stomp 和 WebSocket

STOMP 即 Simple（or Streaming）Text Orientated Messaging Protocol，简单（流）文本定向消息协议，它提供了一个可互操作的连接格式，允许 STOMP 客户端与任意 STOMP 消息代理（Broker）进行交互。STOMP 协议由于设计简单，易于开发客户端，因此在多种语言和多种平台上得到了广泛的应用。

STOMP 协议并不是为 WebSocket 设计的，它是属于消息队列的一种协议，它和 Amqp、Jms 同级。只不过由于它的简单性恰巧可以用于定义 WebSocket 的消息体格式。可以这么理解，WebSocket 结合 Stomp 子协议段，来让客户端和服务端在通信上定义的消息内容达成一致。

STOMP 服务端

STOMP 服务端被设计为客户端可以向其发送消息的一组目标地址。STOMP 协议并没有规定目标地址的格式，它由使用协议的应用自己来定义。例如，/topic/a、/queue/a、queue-a 对于 STOMP 协议来说都是正确的。应用可以自己规定不同的格式以此来表明不同格式代表的含义。比如应用自己可以定义以 /topic 打头的为发布订阅模式，消息会被所有消费者客户端收到，以 /user 开头的为点对点模式，只会被一个消费者客户端收到。

STOMP 客户端

对于 STOMP 协议来说，客户端会扮演下列两种角色的任意一种：

- 生产者，通过 SEND 帧发送消息到指定的地址；
- 消费者，通过发送 SUBSCRIBE 帧到已知地址来进行消息订阅，而当生产者发送消息到这个订阅地址后，订阅该地址的其他消费者会受到通过 MESSAGE 帧收到该消息。

实际上，WebSocket 结合 STOMP 相当于构建了一个消息分发队列，客户端可以在上述两个角色间转换，订阅机制保证了一个客户端消息可以通过服务器广播到多个其他客户端，作为生产者，又可以通过服务器来发送点对点消息。

- 引入依赖

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-websocket</artifactId>
</dependency>
```